

# Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by 昂奕 化学与分子工程学院

说明:

- 1) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。
- 2) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 3) 如果不能在截止前提交作业，请写明原因。

## 1. 题目

### LuoguP1255 数楼梯

dp, bfs, <https://www.luogu.com.cn/problem/P1255>

思路: dp

代码:

```
n=int(input())
stairs=[1,1]+[0]*n
for i in range(2,n+1):
    stairs[i]=stairs[i-1]+stairs[i-2]
print(stairs[n])
```

代码运行截图（至少包含有"Accepted"）

测试点信息 源代码

#### 测试点信息

#1 AC 17ms/4.93MB	#2 AC 15ms/3.71MB	#3 AC 16ms/3.72MB	#4 AC 15ms/3.60MB	#5 AC 16ms/3.70MB	#6 AC 15ms/3.61MB
#7 AC 16ms/3.75MB	#8 AC 16ms/3.78MB	#9 AC 16ms/4.09MB	#10 AC 17ms/5.04MB		



所属题目	P1255 数楼梯
评测状态	Accepted
提交时间	2024-11-28 13:44:04

### 27528: 跳台阶

dp, <http://cs101.openjudge.cn/practice/27528/>

思路: 上n级台阶有以下方法:

for i in range(n):

上i级台阶

然后一步跨到第n级

所以状态转移方程就是dp[n]=dp[0]+dp[1]+...+dp[n-1].其中dp[0]=1,dp[1]=1

代码:

```

n=int(input())
dp=[0]*(n+1)
dp[1]=1
for i in range(1,n+1):
    dp[i]=sum(dp[j] for j in range(i))+1
print(dp[n])

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

n=int(input())
dp=[0]*(n+1)
dp[1]=1
for i in range(1,n+1):
    dp[i]=sum(dp[j] for j in range(i))+1
print(dp[n])

```

## 474D. Flowers

dp, <https://codeforces.com/problemset/problem/474/D>

思路:

最开始想到的是数学方法(用组合数),但是组合数算起来很费时间(下面这一行是超时后感想)

dp超时了...用pypy甚至更慢(因为数据量小的时候pypy启动时间占主导)

然后换了一种dp思路

dp,但是进行dp的时候遇到累加可以用前缀和,省时间

进行dp的时候遇到累加可以用前缀和,省时间

dp的时候遇到累加可以用前缀和,省时间

遇到累加可以用前缀和,省时间

可以用前缀和,省时间

前缀和,省时间

,省时间(超时了几十次的血的教训...哭)

不能到最后一步再取模,会算出几万位的大整数,会爆内存...(不能仗着python能算大整数就为所欲为)

取模对加减乘是没有影响的,可以放心取.但除法不能先取模再算

另:学到了一个骗取(不是)CF测试数据的办法:

if 第i组数据WA:

将输出的代码增加一个判断:

if 第i组数据:

print(输入的数据)

代码:

```

import sys
MAXN=1000000007
def dp(maxin):
    dp = [1] * (100099)
    ans = list(range(k))+[k+1] # 存前缀和的, 到时候输出就直接ans[b]-ans[a-1]
    anstmp = ans[-1]
    dp[k]=2
    for n in range(k+1,maxin+1):
        dp[n]=(2*dp[n-k]+ans[n-k-1]-ans[n-2*k])%MAXN if (n-2*k) >=0 else (2*dp[n-k]+ans[n-k-1]+1)%MAXN
        anstmp+=dp[n]
        anstmp%=MAXN
        ans.append(anstmp%MAXN)
    return ans
pyin=sys.stdin.readlines()

```

```
t,k=map(int,pyin[0].split())
maxin=0
pyin.pop(0)
for line in pyin:
    a,b=map(int,line.split())
    maxin=max(a,b,maxin)
ans=dp(maxin)
for line in pyin:
    a,b=map(int,line.split())
    print((ans[b]-ans[a-1])%MAXN)
```

代码运行截图（至少包含有"Accepted"）

#	When	Who	Problem	Lang	Verdict	Time	Memory
<a href="#">293966681</a>	Nov/30/2024 17:06 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Accepted	483 ms	16900 KB
<a href="#">293966537</a>	Nov/30/2024 17:05 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 34	1500 ms	262100 KB
<a href="#">293966475</a>	Nov/30/2024 17:05 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Runtime error on test 1	46 ms	0 KB
<a href="#">293964964</a>	Nov/30/2024 16:51 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 1	546 ms	262100 KB
<a href="#">293964426</a>	Nov/30/2024 16:47 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 25	1421 ms	262100 KB
<a href="#">293963849</a>	Nov/30/2024 16:42 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 25	1405 ms	262100 KB
<a href="#">293963233</a>	Nov/30/2024 16:36 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 8	515 ms	262100 KB
<a href="#">293945178</a>	Nov/30/2024 13:59 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 1	46 ms	300 KB
<a href="#">293943665</a>	Nov/30/2024 13:43 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 3	1500 ms	6300 KB
<a href="#">293942091</a>	Nov/30/2024 13:27 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	77 ms	5600 KB
<a href="#">293617310</a>	Nov/28/2024 16:59 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	PyPy 3-64	Time limit exceeded on test 5	1500 ms	200300 KB
<a href="#">293616991</a>	Nov/28/2024 16:57 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	PyPy 3-64	Memory limit exceeded on test 1	77 ms	262100 KB
<a href="#">293616873</a>	Nov/28/2024 16:56 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 8	1500 ms	129200 KB
<a href="#">293616774</a>	Nov/28/2024 16:55 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 2	1500 ms	31700 KB
<a href="#">293616118</a>	Nov/28/2024 16:49 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 5	1500 ms	116800 KB
<a href="#">293614715</a>	Nov/28/2024 16:37 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 1	1500 ms	72700 KB
<a href="#">293614386</a>	Nov/28/2024 16:34 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Runtime error on test 1	46 ms	1300 KB
<a href="#">293613335</a>	Nov/28/2024 16:25 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	PyPy 3-64	Time limit exceeded on test 5	1500 ms	162400 KB
<a href="#">293612840</a>	Nov/28/2024 16:23 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	PyPy 3-64	Time limit exceeded on test 5	1500 ms	156100 KB
<a href="#">293612432</a>	Nov/28/2024 16:20 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Time limit exceeded on test 8	1500 ms	127300 KB
<a href="#">293611706</a>	Nov/28/2024 16:16 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 8	1124 ms	262100 KB
<a href="#">293606429</a>	Nov/28/2024 15:41 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	202 ms	14100 KB
<a href="#">293606231</a>	Nov/28/2024 15:40 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	202 ms	14100 KB
<a href="#">293606052</a>	Nov/28/2024 15:39 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	233 ms	14100 KB
<a href="#">293605893</a>	Nov/28/2024 15:37 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	249 ms	14500 KB
<a href="#">293603071</a>	Nov/28/2024 15:14 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	296 ms	17800 KB
<a href="#">293603029</a>	Nov/28/2024 15:14 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 1	77 ms	4400 KB
<a href="#">293601739</a>	Nov/28/2024 15:03 <sup>UTC+8</sup>	_12_	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	280 ms	17400 KB

<a href="#">93603071</a>	Nov/28/2024 15:14 <sup>UTC+8</sup>	_ -12- _	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	296 ms	17800 KB
<a href="#">93603029</a>	Nov/28/2024 15:14 <sup>UTC+8</sup>	_ -12- _	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 1	77 ms	4400 KB
<a href="#">93601739</a>	Nov/28/2024 15:03 <sup>UTC+8</sup>	_ -12- _	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	280 ms	17400 KB
<a href="#">93601204</a>	Nov/28/2024 15:00 <sup>UTC+8</sup>	_ -12- _	<a href="#">D - Flowers</a>	Python 3	Wrong answer on test 2	281 ms	17800 KB
<a href="#">93600872</a>	Nov/28/2024 14:58 <sup>UTC+8</sup>	_ -12- _	<a href="#">D - Flowers</a>	Python 3	Memory limit exceeded on test 2	562 ms	262100 KB

## LeetCode5.最长回文子串

dp, two pointers, string, <https://leetcode.cn/problems/longest-palindromic-substring/>

思路：子串,是(((连续)))的非空子序列.连续啊啊啊啊啊啊啊啊我就说这么久怎么想不出来  
然后就很平凡了  
代码：

```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        s=list(s)
        maxl=1
        maxstr=[s[0]]
        for i in range(1,len(s)):
            for j in range(1,min(i+1,len(s)-i)):
                if j < len(s)-i and s[i-j]==s[i+j]:
                    if 2*j+1>maxl:
                        maxl=2*j+1
                        maxstr=s[i-j:i+j+1]
                else:break
            for j in range(1,min(i+1,len(s)-i+1)):
                if s[i-j]==s[i+j-1]:
                    if 2*j>maxl:
                        maxl=2*j
                        maxstr=s[i-j:i+j]
                else:break
        return (''.join(maxstr))
```

代码运行截图 （至少包含有"Accepted"）

通过

12 提交于 2024.12.01 10:56

官方题解

写题解

### ⌚ 执行用时分布



291 ms | 击败 71.54% 🏆

💡 复杂度分析

### 💰 消耗内存分布

17.37 MB | 击败 52.05% 🏆

## 12029: 水淹七军

bfs, dfs, <http://cs101.openjudge.cn/practice/12029/>

思路:

处理输入比bfs难...

两个代码,第一个是OJ的,第二个是高度相同也能淹的

某点淹了水之后调整高度

地图里的点进行如下判断:

如果四周都被淹,则这个点不再拥有淹没其他点的能力,判断为dead,除非有比它更高的水淹到它

如果被淹,判断为drowned

入队条件为:待入队的点高度比当前点高度低或者相等,但如果高度相等且待入队的点已死就不再入队,这样淹过的点也能再淹,

代码:

```
from collections import deque
import sys
MAXN=1111
def bfs(x,y):
    global slb,mat
    q=deque([(x,y)])
    d=[(-1,0),(0,1),(1,0),(0,-1)]
    while q:
        x,y=q.popleft()
        for dx,dy in d:
            nx,ny=x+dx,y+dy
            if 1<=nx<=m and 1<=ny<=n \
            and mat[nx][ny]<mat[x][y]:
                q.append((nx,ny))
                mat[nx][ny]=mat[x][y]
                if (nx,ny)==slb:
                    return('Yes')
    return('No')
lines=list(sys.stdin.read().split())

t=int(lines[0])
id=1
for _ in range(t):
    m,n=map(int,(lines[id],lines[id+1]))
    mat=[ [MAXN]*(n+2)]
    id+=2
    for i in range(m):
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
from collections import deque
import sys
MAXN=1111
def bfs(x,y):
    global slb,mat
    q=deque([(x,y)])
    d=[(-1,0),(0,1),(1,0),(0,-1)]
    while q:
        x,y=q.popleft()
        for dx,dy in d:
            nx,ny=x+dx,y+dy
            if 1<=nx<=m and 1<=ny<=n \
            and mat[nx][ny]<mat[x][y]:
                q.append((nx,ny))
                mat[nx][ny]=mat[x][y]
                if (nx,ny)==slb:
                    return('Yes')
        return('No')
lines=list(sys.stdin.read().split())

t=int(lines[0])
id=1
for _ in range(t):
    m,n=map(int,(lines[id],lines[id+1]))
    mat=[[MAXN]*(n+2)]
    id+=2
    for i in range(m):
```

基本信息

#: 47461768  
题目: 12029  
提交人: 24n2400011782  
内存: 7924kB  
时间: 50ms  
语言: Python3  
提交时间: 2024-11-29 15:41:37

## 02802: 小游戏

bfs, <http://cs101.openjudge.cn/practice/02802/>

思路:

我编程序哪里牛了...靠这个赚钱...想都别想...

细节一团糟...

问题1:如果可以走到保护圈上,那么可以去某位置探路的条件为 $0 \leq nx < w+1$  &&  $0 \leq ny \leq h+1$ 而不是 $1 \leq$

问题2:如果题目里x表示横坐标,y表示纵坐标,矩阵里的点应用mat[y][x]表示

问题3:使用sys.stdin时有一大堆问题:

3-1 对于矩阵的每一行输入都要rstrip('\n'),不然会导致保护圈上的元素被换行符挤走

3-2 一次性读入数据之后用一个指针id指着即将读取的行或者元素来读,这种方法虽然straightforward但是不会错而且不费时不费内存

问题4:对于初始和终止条件的考虑,想好这些条件再写比先写个大概然后各种缝缝补补舒服得多

代码:

```
from collections import deque
import sys
from copy import deepcopy
def bfs(x,y):
    matc=deepcopy(mat)
    q=deque([(x,y,0,0)])#分别是x,y,线段数,来时的方向
    d=[(-1,0),(0,1),(0,-1),(1,0)]
    ans=99999
    ss=[[99999]*(w+2) for _ in range(h+2)]
    while q:
        x,y,s,dr=q.popleft()
        for cntdr in range(len(d)):
            dx,dy=d[cntdr]
            nx,ny=x+dx,y+dy

            if 0<=nx<=w+1 and 0<=ny<=h+1:
                #这个ans存的是当前已经到终点的路的最小线段数,如果走到了哪个位置发现s已经>这个ans了,就不要再走这个位置了
                if (nx,ny)==(x2,y2):
```

```
ans = min(s+1 if cntdr!=dr or s==0 else s,ans)
ss[ny][nx]=ans
if matc[ny][nx]==' ' or (matc[ny][nx]=='V' and ss[y][x]>=s):
    if cntdr==dr and s<=ans and s!=0:
        q.append((nx,ny,s,cntdr))
    elif s+1<=ans:
        q.append((nx,ny,s+1,cntdr))
matc[y][x]='V'
cc[v][v]=min(cc[v][v],c)
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
from collections import deque
import sys
from copy import deepcopy
def bfs(x,y):
    matc=deepcopy(mat)
    q=deque([(x,y,0,0)]) #分别是x,y,线段数,来时的方向
    d=[(-1,0),(0,1),(0,-1),(1,0)]
    ans=99999
    ss=[[99999]*(w+2) for _ in range(h+2)]
    while q:
        #print(q)
        x,y,s,dr=q.popleft()
        #print(f'正在{x},{y}')
        for cntdr in range(len(d)):
            dx,dy=d[cntdr]
            nx,ny=x+dx,y+dy

            if 0<=nx<=w+1 and 0<=ny<=h+1:
                #print(f'正在探{nx},{ny},该位置是{mat[ny][nx]}')
                #这个ans存的是当前已经到终点的路的最小线段数,如果走到了哪个位置发
                if (nx,ny)==(x2,y2):
                    #print(f'看到终点,当前位置:{x},{y},终点在{cntdr}方,来时方
                    ans = min(s+1 if cntdr!=dr or s==0 else s,ans)
                    #print('ans:',ans)
                    ss[ny][nx]=ans
                    #print(f'matc[{ny}][{nx}]=={matc[ny][nx]}')
                    if matc[ny][nx]==' ' or (matc[ny][nx]=='V' and ss[y][x]>=s):
```

基本信息

#: 47484365  
题目: 02802  
提交人: 24n2400011782  
内存: 3900kB  
时间: 110ms  
语言: Python3  
提交时间: 2024-11-30 18:23:01

## 2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

作业题不简单...细节很有问题.

可能是做得最痛苦的一次呜呜呜呜

另外sys.stdin是可以本地调试的,按Ctrl+D可以EOF

注意stdin的结果里可能从某些角落里冒出来的换行符!!!!